VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Použití n-gramů pro detekci spamu

Using N-grams to Detect Spam

2014 Tomáš Tománek

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Tomáš Tománek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Použití n-gramů pro detekci spamu
Using N-grams to Detect Spam

Zásady pro vypracování:

Metody využívající n-gramy jsou v dnešní době aplikovány na různé problémy v oblasti vyhledávání a analýzy textových dokumentů. Cílem této práce je návrh a implementace metody pro detekci spamu pomocí n-gramů.

1. Vypracujte rešerši týkající se použití n-gramů při zpracování a analýze textových dokumentů, zaměřte se na detekci spamu.
2. Navrhněte a implementujte aplikaci pro detekci spamu využívající n-gramy.
3. Proveďte testování a výsledky porovnejte s dalšími metodami.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Václav Bašniar**

Datum zadání: 16.11.2012
Datum odevzdání: 07.05.2014

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * Fakulta elektrotechniky a informatiky * OSTRAVA *

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.*

V Ostravě 16. dubna 2014 .....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 16. dubna 2014 .....

Rád by som sa poďakoval všetkým, ktorí ma podporovali a bez ktorých by táto práca nikdy nevznikla. Hlavne mojej priateľke a vedúcemu bakalárskej práce p. Ing. Václavovi Bašniarovi za odbornú pomoc a cenné rady, ktoré mi poskytol pri vypracovávaní tejto práce.

Abstrakt

Cieľom bakalárskej práce je navrhnúť aplikáciu na detekciu spamu pomocou metódy extrakcie n-gramov. V práci sú popísané dôležité pojmy týkajúce sa extrakcie n-gramov pri spracovaní textov, takisto aj samotný pojem spam a spôsoby detekcie spamu. Na záver sú získané výsledky navrhnutej aplikácie detekcie spamu porovnávané s výsledkami doterajších spôsobov detekcie spamu.

Klíčová slova: n-gram, detekcia spamu, spam

Abstract

The goal of this bachelor thesis is to design the application for spam detection using the method of extracting n-grams. The work describes the essential terms and concepts of extracting n-grams in text processing, and the concept of spam itself and spam detection methods as well. In the summary, the results obtained by the proposed application of spam detection are compared with the results of existing methods of spam detection.

Keywords: n-gram, spam detection, spam

Seznam použitých zkratek a symbolů

SMTP	–	Simple Mail Transfer Protocol
DNS	–	Domain Name Server
XML	–	Extensible Markup Language
SSL	–	Secure Sockets Layer
LINQ	–	Language-Integrated Query
AI	–	Artificial Intelligence

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Pre dnešnú dobu je špecifická vysoká informatizácia. Skoro každá domácnosť má zavedené pripojenie na internet. Je ohromujúce, ako sa väčšina komunikácie, obchodných ponúk a objednávok presunula do elektronickej podoby. Tento krok je logický, pretože nás táto komunikácia nič nestojí a vybavíme ju okamžite a odkiaľkoľvek. Dnes už má prakticky každý svoju e-mailovú schránku. Prináša to veľa výhod ako rýchlosť, úspora času či úspora finančných prostriedkov, ale tak isto aj isté riziká. Jedným z nich je nevyžiadaná pošta. Ide o podvodné správy ako napríklad "Vyhrali ste v lotérii...", Nigérijské listy, hoax - poplašné falošné správy alebo správy obsahujúce počítačový vírus, ktorý môže poškodiť používateľské dáta, získať dôležité heslá, napríklad do internet bankingu alebo vo výnimočných prípadoch poškodiť osobný počítač.

Cieľom mojej práce je napísať program, ktorý bude detekovať nevyžiadanú poštu pomocou technológie extrakcie n-gramov z tela správy. Vo svojej podstate je to úloha spracovania a kategorizácie textu, preto sa používajú rovnaké metódy na zisťovanie podobností textov. Pomocou n-gramov dokážeme zachytiť slovné spojenia i vetné presahy. Práve preto je táto metóda pri kategorizácii dlhších textov veľmi účinná. Zaujímavé bude však sledovať, ako sa metóda bude vyvíjať pri kratších textových dokumentoch, akými sú častokrát e-mailové správy.

2 N-gram

S pojmom n-gram sa stretávame v bežnom živote už takmer na každom kroku, niekedy si to ani sami neuvedomujeme. N-gram je definovaný ako sekvencia n symbolov (napríklad slov, hlások atď.) [31, 39]. N-gramy majú široké využitie v rôznych oblastiach. V oblasti hudby [36], genetiky [37] či dokonca videohier [38]. Najväčšie uplatnenie však našli pri spracovaní textu. Ide o bežné veci, ktoré nám každodenne uľahčujú život a častokrát si to ani neuvedomujeme.

Pri hudbe dokážeme na základe sledu tónov určiť skladbu alebo hudobný štýl. V genetike dokážeme na základe DNA zostaviť vzorec jedinečný napríklad pre určitý genotyp ľudí alebo ľudí s rovnakou vrodenou chorobou. Vo videohrách sa postupnosti a kombinácie prvkov vyskytujú pomerne často. Najtypickejším príkladom sú takzvané bojové hry, kde sú jednotlivé kombá kombináciou tlačidiel v krátkom časovom intervale. Tieto kombinácie sú vo svojej podstate n-gramy rôznej dĺžky, ktoré vyvolajú požadovanú akciu. AI pomocou predikcie dokáže predvídať na základe začatej sekvencie ťahov hráčov úmysel a reagovať naň. Pomocou citlivosti predikcie dokážeme nastaviť aj obtiažnosť. Medzi najznámejšie hry tohoto typu patria série Tekken, Mortal Kombat či Street fighter [42].

Veľmi významným sa stalo odhaľovanie plagiátov [34, 35]. S rozvojom internetu sa začalo čoraz viac prác, dokumentov a štúdií objavovať online, a tak sa stali oveľa dostupnejšie a ľahšie vyhľadateľné. Na jednej strane je takáto dostupnosť dobrá pre štúdium, ale na strane druhej sa tieto informácie začali zneužívať a ľudia si často uľahčovali prácu kopírovaním a prisvojovaním si cudzích myšlienok. Na odhaľovanie plagiátorstva na základe n-gramov sa najčastejšie používajú slovné n-gramy. Na základe slovných spojení dokážeme ľahko detekovať rovnaké časti dokumentov aj vetné presahy, čo umožní jednoznačne detekovať, na koľko je práca plagiátom.

Ďalším, taktiež veľmi dôležitým a zaujímavým využitím n-gramov, je detekcia jazyka a kategorizácia textových dokumentov. Pre rozoznanie jazyka dokumentu sa používajú n-gramy zložené z jednotlivých písmen. Každý jazyk má svoje špecifické vlastnosti, na základe ktorých ho vieme takmer jednoznačne identifikovať [39]. Pri kategorizácii textov tak isto zohrala dôležitú úlohu informatizácia a hlavne digitalizácia knižníc. Jednotlivé diela vieme na základe n-gramov jednoznačne zaradiť do kategórií. Jednotlivé kategórie musíme najskôr špecifikovať. Príkladom kategórií sú napríklad román, krimi, horor, sci-fi a podobné delenie literatúry.

Všetky spôsoby využitia n-gramov majú jednu vlastnosť, ktorú sa pokúsime aplikovať aj v našej práci. Touto vlastnosťou je, že na zaradenie alebo identifikáciu, ktorú požadujeme, potrebujeme definovať pravidlá, ako napríklad pri rozpoznávaní jazyka, kategorizácii textov, prípadne mať k dispozícii dostatočne veľkú databázu prvkov, ako napríklad pri detekcii plagiátov, nájdenie skladby v hudbe. Cieľom našej práce je pokúsiť sa na základe n-gramov detekovať spam. Na detekciu spamu teda využijeme čo možno najväčšiu databázu nevyžiadaných správ. Využijeme skutočnosť, že spamy bývajú zväčša rovnaké. Budeme teda porovnávať požadovanú správu s databázou a budeme hľadať, či je niektorý zo spamov v databáze podobný a do akej miery. Miera podobnosti bude vyjadrená pomerom zhodných n-gramov porovnávanej správy s podobnou správou z da-

tabázy a všetkých n-gramov podobnej správy z databázy. Ak bude porovnávaná správa podobná minimálne jednej správe v databáze, je veľmi pravdepodobné, že táto správa bude spamom.

3 Dátové štruktúry na ukladanie n-gramov

Na ukladanie n-gramov sa používajú klasické dátové štrukúry, najčastejšie pole. Pri ukladaní n-gramov sa v začiatkoch ukladali úplne všetky n-gramy a ich početnosť sa získala spočítaním všetkých rovnakých n-gramov v poli. Táto metóda však zbytočne ukladala rovnaké n-gramy do poľa. Vo výskume [41] išlo o 5,1% - 56,3% zhodných n-gramov v korpuse. Práve z tohoto dôvodu sa hľadali efektívnejšie riešenia na ukladanie n-gramov.

Vhodným riešením boli dátové štruktúry, ktoré používajú dvojicu $key, value$ na ukladanie dát. Zvyčajne ich nazývame aj mapy. Keďže nie je možné mať dva rovnaké kľúče, vylúčime tak nežiadúce duplicity. Nájsť vhodnú štrukúru však nie je taká jednoduchá úloha, ako sa na prvý pohľad môže zdať, hlavne keď potrebujeme početnosť uložených dát.

Efektívne ukladanie n-gramov do takýchto dátových štruktúr prebieha nasledovne. Samotná postupnosť prvkov je kľúčom, pretože potrebujeme zaistiť jej jedinečnosť. Počet výskytov, váhu n-gramu alebo napríklad index dokumentu, v ktorom sa daný n-gram nachádza, ukladáme do hodnoty, teda value. Hodnotou môže byť, samozrejme, aj ďalšia dátová štruktúra, ktorá dopĺňa kľúč.

Hľadanie prvkov v poli prvkov tvorených z dvojích je často náročné na výpočtový výkon, pretože je potrebné hľadaný prvok porovnať so všetkými prvkami v poli. Aby sme sa takémuto prístupu vyhli, môžeme použiť binárne vyhľadávanie. Musíme mať však na pamäti, že v tom prípade musíme mať toto pole zoradené.

Preto boli vynájdené dátové štruktúry, ktoré sú oveľa účinnejšie pri vkladaní prvkov či vyhľadávaní prvkov.

3.1 Hašovacia tabuľka

Hašovacia tabuľka je dátová štruktúra, ktorá združuje kľúče a hodnoty. Kľúč je v hašovacej tabuľke prevádzaný na index pomocou hašovacej funkcie. Primárnou funkciou je vyhľadávanie, ktoré je vďaka indexom veľmi efektívne a rýchle.

Vzhľadom k tomu, že v jednom elemente je možné pri hašovacích tabuľkách uložiť iba jednu položku, musíme počítať aj s riešením kolízií. Na riešenie kolízií máme dve možnosti. Prvou možnosťou je, že sa zhodný záznam uloží do nasledujúceho voľného elementu. Druhou možnosťou je, že budeme položky ukladať do spájaného zoznamu [4].

3.2 B+ strom

Štruktúra B+ stromu je založená na B strome, umožňujúca rýchle vyhľadávanie, vkladanie či mazanie dát. Na rozdiel od B stromu sú dáta ukladané iba v listoch. Kľúče sú uložené aj vo vnútorných uzloch a v koreni.

Požiadavky na B+ strom môžeme charakterizovať v štyroch bodoch:

- koreň má maximálne N potomkov,
- každý uzol s výnimkou koreňa má maximálne N a minimálne N/2 potomkov,

- dáta sú uložené len v listoch,
- všetky listy sú na rovnakej úrovni.

Splnením týchto požiadaviek dostávame striktne vyváženú dátovú štruktúru pre ukladanie dát. N-gramy sú ukladané v B+ stromoch ako sekvencie. Jedinou nevýhodou je, že pri vyhľadávaní je nutné porovnávať rovnaké prefixy sekvencií, čo algoritmus spomaľuje [4].

3.3 Ternárny AVL strom

Binárny vyhľadávací strom je dátová štruktúra pozostávajúca z vrcholov, pričom každý obsahuje hodnotu a hranu. V binárnom strome existuje práve jeden koreň a každý uzol má práve dvoch potomkov, pričom jeden z potomkov obsahuje väčšiu a jeden menšiu hodnotu akú má ich rodič.

Ak chceme nájsť prvok, začíname od koreňa a jednoduchým porovnaním nájdeme požadovaný výsledok. Vďaka tomu dosahuje vyhľadávanie zložitosť $O(log_2 n)$, kde n je počet prvkov uložených v strome.

Pri vytváraní stromovej štruktúry môže nastať situácia, kedy sa vytvorí spojený zoznam namiesto stromu. Takto vytvorený strom má zložitosť hľadania v priemere $O(n/2)$. Tento problém môžeme vyriešiť použitím samovyvažovacieho stromu, akým je napríklad AVL strom. Ide o binárny vyhľadávací strom, ktorý ešte navyše spĺňa podmienku, že rozdiel dvoch podstromov každého uzla je najviac jedna. Pridávanie nového prvku môže vyžadovať vyváženie stromu jednou alebo viacerými rotáciami stromu [4].

Tento prístup síce zvyšuje časovú náročnosť pri vkladaní a mazaní uzlov, ale zaisťuje efektívnejšie vyhľadávanie pri vkladaní neusporiadaných dát. Ukladanie n-gramov môže byť riešené presne tak ako u B+ stromov. To znamená, že každý uzol bude obsahovať celý n-gram. Stále ale nastáva problém prechádzania rovnakých predpôn pri vyhľadávaní. Ternárny AVL strom je upravená verzia AVL stromu, ktorá okrem dvoch odkazov na potomkov obsahuje ešte tretí údaj a tým je odkaz na najbližší podstrom. Tento odkaz obsahuje iba časť kľúča bez prefixu, ktorý určuje vyšší podstrom.

3.4 Hybridný AVL strom

Meraním bolo zistené [5], že použitie ternárneho AVL stromu pri hĺbke viac ako štyri je neefektívne. Z toho dôvodu sa používa v tomto prípade hybridný AVL strom. Hybridný AVL strom rieši problém nasledovným spôsobom: namiesto jedného AVL stromu s hĺbkou väčšou ako štyri zostaví hybridný AVL strom. Do koreňa je uložená hašovacia tabuľka obsahujúca korene menších ternárnych AVL stromov, na ktoré je rozdelený pôvodný strom [6].

3.5 Dvojitý ternárny AVL strom

V prípade, že použijeme n-gramy ako n-tice slov, nastane značná redundancia, vďaka ktorej výrazne vzrastie spotreba operačnej pamäte. Ak n-gram rozdelíme na slová, každé slovo môže byť indexované samostatne. Indexovaním sa rozumie prevod z textového tvaru slova na číselnú hodnotu [3]. V texte sa slová často opakujú, preto je vhodné mať pre rovnaké slová rovnaké indexy. Tým sa vyhneme redundancii.

Dvojitý ternárny AVL strom je zložený z dvoch ternárnych AVL stromov. V prvom strome sú ukladané všetky slová. V druhom strome sú ukladané n-gramy zložené z indexov obsahujúcich slov. Algoritmus najskôr zistí, či sa v prvom strome nachádzajú všetky slová n-gramu, ak áno, prejde na druhý strom.

4 Algoritmy na extrakciu n-gramov

4.1 Nagao 94

Algoritmus bol navrhnutý špeciálne pre japonské texty. Japonská abeceda obsahuje okolo 4000 znakov, pričom každý znak reprezentuje jedno slovo. Tento algoritmus nám reprezentuje frekvenciu výskytu postupností znakov. Pozostáva z dvoch fáz. V prvej fáze je vstupom text reprezentovaný ako textový reťazec skladajúci sa zo slov a prázdnych znakov. Do prvého poľa sa uloží tento text. V druhom poli sa nachádzajú pointery na podreťazce vstupného textu v prvom poli.

Podreťazec, na ktorý ukazuje pointer $n - 1$, je definovaný ako skladba znakov počínajúc n-tou pozíciou až po koniec textového reťazca. Následne sa zoradí pole pointerov takým spôsobom, že zhodné podreťazce sa nachádzajú pod sebou. Druhé pole zostáva nezmenené.

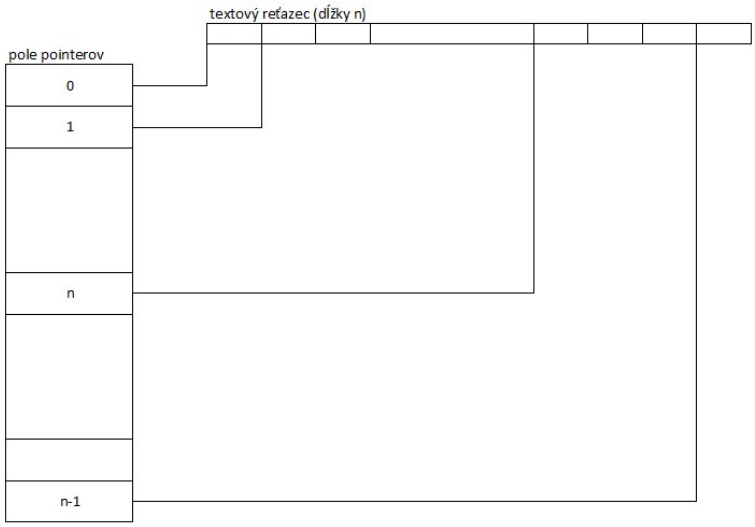
Následne sa v zoradenom poli porovnajú každé dva susedné podreťazce a vypočíta sa veľkosť prefixu, ktorá je zhodná pre oba porovnávané podreťazce. Výsledok sa zapíše do tabuľky prefixov.

Výsledkom druhej fázy je početnosť n-gramov. Na výpočet početnosti sa používa práve tabuľka prefixov znakov získaná v prvej fáze algoritmu. Po tom, ako je zvolená veľkosť n-gramu, sa prechádza pole pointerov nasledovným spôsobom. Načíta sa prvých n znakov prvého slova a kontroluje sa číslo v tabuľke prefixov. V prípade, že je toto číslo väčšie alebo rovné n, znamená to, že nasledujúce slovo má minimálne n spoločných prefixových znakov s prvým slovom. Algoritmus ďalej kontroluje ďalší záznam v tabuľke a porovnáva sa s n. Postup sa opakuje, pokiaľ nie je číslo uložené v tabuľke prefixov menšie ako n. Frekvenciu výskytu n znakov z prvého slova dostaneme ako počet prejdených slov od prvého slova. Algoritmus sa opakuje pre všetky slová textového reťazca [1].

4.2 Lempel-Ziv-Welch algoritmus

Tento algoritmus má základ v Lempel-Ziv-Welch algoritme bezstratovej kompresie dát. Funguje tak, že sa postupne vytvára kódovacia tabuľka tvorená slovami nachádzajúcimi sa v zakódovanom texte. Vstup je v tejto tabuľke mapovaný na slová, ktoré majú pevne stanovenú dĺžku. Najčastejšie býva inicializovaná na základe všetkých znakov ASCII tabuľky. Následne algoritmus sekvenčne prehľadá vstupný textový reťazec a do tabuľky zapíše každé unikátne dvojznakové slovo ako spojenie kódu a vzoru. Ak uloží všetky dvojznakové slová, je na výstup zadaný kód prvého znaku na vstupe. Algoritmus pokračuje vo svojej činnosti, v prípade, že narazí na už zakódované slovo, tak na výstup odošle už zakódovaný znak s prvým znakom slova.

Príklad takéhoto algoritmu je znázornený vývojovým diagramom na nasledujúcom obrázku. Algoritmus využíva vyššie popísaný princíp LZW na extrakciu n-gramov zo zadaného textu. Na uloženie extrahovaných vzoriek používa binárny strom. Vzorkami sú v tomto algoritme slová [2].



Obrázek 1: Nagao 94 algoritmus

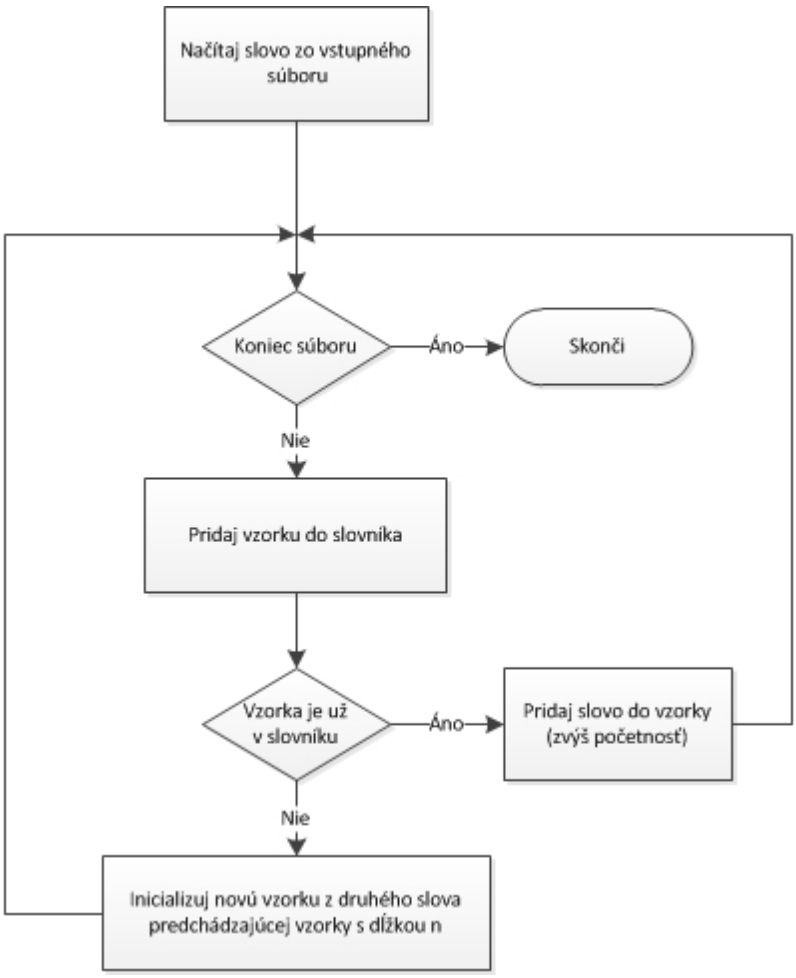
4.3 Sufixové pole

Sufixové pole je dátová štruktúra, ktorá sa často využíva na získanie n-gramov a ich početnosti z textu. Sufixové pole pozostáva z N sufixov, ktoré sú zoradené abecedne. Sufix $s[i]$ je reťazec začínajúci na i-tej pozícii vo vstupnom reťazci a pokračuje až do konca reťazca.

Frekvencia termov sa počíta tak, že odčítame index prvého výskytu termu a index posledného výskytu termu a pripočítame číslo jeden. Frekvencie sa ukladajú do LCP (longest common prefix) poľa. Po naplnení tohoto poľa je dobré vytvoriť štruktúru tzv. n-gram deskriptorov obsahujúcu dvojice (koncová pozícia, frekvencia). Na základe uloženia deskriptorov do štruktúr je možné následne vykonávať užitočné operácie nad n-gramami. Ak chceme nájsť na výstupe n-gramy s nízkou frekvenciou, použijeme prvú funkciu a vytriedime menej frekventované deskriptory. Druhou operáciou je zoradenie deskriptorov na základe koncovej pozície. Výsledkom je abecedné zoradenie deskriptorov. Výstup v podobe n-gramov dostaneme extrakciou z desktriptorov pomocou koncovej pozície a frekvencie, ktoré sú uložené v deskriptore, a požadovanej dĺžky n-gramov [7].

4.4 Sufixový strom

Sufixový strom je tak isto ako sufixové pole dátová štruktúra. Vrchol sufixového stromu tvorí časť podreťazca, ktorý dostaneme ako zreťazenie hrán na ceste z koreňa až po list. Metódy extrakcie n-gramov pomocou sufixových stromov a polí sú takmer podobné, rozdielom je iba dátová štruktúra, v ktorej sú uložené jednotlivé slová vstupného reťazca.



Obrázek 2: Reprezentácia LZV algoritmu

V uzloch stromu nie sú uložené kompletné slová, ale každý synovský uzol ukazuje na podreťazec slov v poli obsahujúcom vstupný textový reťazec. Následný postup extrakcie je zhodný s postupom, kde sa používa sufixové pole.

Na extrakciu slovných n-gramov sú veľmi často využívané práve metódy sufixového poľa a sufixového stromu. Ďalšie využitie sufixových stromov:

- nájdenie všetkých výskytov slova v zadaných reťazcoch,
- určenie, či je dané slovo podreťazcom iného slova,
- vyhľadávanie dvojíc opakovaných výskytov,
- vyhľadávanie maximálnych opakovaní,
- vyhľadávanie najdlhšieho palindrómu v texte,
- kompresia dát,
- identifikácia osôb pomocou DNA.

Jedinou nevýhodou sufixového stromu je jeho priestorová náročnosť.

4.5 Invertovaný index

Pre potreby spracovania prirodzeného jazyka sa často používa aj invertovaný index. Táto štruktúra je veľmi účinná pri vyhľadávaní podreťazcov v texte. Na vyhľadávanie využíva práve n-gramy. Invertovaný index obsahuje dvojicu:

- zoznam dokumentov, ktoré obsahujú konkrétny term,
- pre každý term vo vstupnom reťazci obsahuje frekvenciu výskytu a pozície, na ktorých sa nachádza.

Invertovaný index nie je najvhodnejším nástrojom na extrakciu n-gramov hlavne pre svoju vyššiu náročnosť a neefektívnosť v prípade dlhších n-gramov. Využitie si ale našiel v už spomínanom vyhľadávaní podreťazcov a bioinformatike. Je nezávislý na spracovávanom jazyku dokumentu, a preto sa často používa pri práci s japonskými a kórejskými textami [9].

4.6 Apriori algoritmus

Úlohou algoritmu je nájdenie všetkých asociačných pravidiel vzhľadom na zadanú množinu. Podpora pravidiel musí byť minimálne taká, aká je zadaná podpora a spoľahlivosť musí byť väčšia ako minimálna zadaná podpora. Algoritmus môžeme rozdeliť na 2 podúlohy:

- nájdenie všetkých frekventovaných množín, ktorých podpora je väčšia ako zadaná podpora,
- z frekventovaných množín vytvoriť asociačné pravidlá so splnenou podmienkou minimálnej spoľahlivosti.

Práca algoritmu spočíva v tom, že pri prvom prechádzaní databázy si nájde všetky jednoprvkové množiny vyhovujúce minimálne podpore. V nasledujúcom kroku vygeneruje kandidátov na dvojprvkové frekventované množiny na základe už vytvorených jednoprvkových. Tento cyklus sa opakuje do chvíle, keď už nie je možné vygenerovať ďalšie vyhovujúce množiny [10].

Pre prácu s n-gramami sa používa modifikácia algoritmu apriori.

5 Spam

Definícia termínu spam sa líši podľa použitého zdroja. Vo všeobecnosti ale ide o nevyžiadanú poštu z jednej správy alebo kolekcie správ poslaných na veľký počet adries posielaných opakovane pri minimálnej zmene obsahu [11]. Taktiež môžeme povedať, že ide o správy písané strojom. Sú automaticky posielané zväčša na veľkú kolekciu adries v čo najkratšom čase s cieľom získať čo najväčší zisk. Spam nemusí byť len obchodného charakteru. Môže ísť aj o reťazové správy, prípadne správy s politickou či náboženskou tematikou.

5.1 História pojmu

Názov spam pochádza pôvodne zo značky amerických konzerv lanšmítu, ktorá sa vyrába už od 30. rokov dodnes. V súčasnosti výrobca trvá na písaní názvu veľkými písmenami, teda SPAM. Za 2. Svetovej vojny a po nej bol tento produkt stále viac a viac rozšírený a menej a menej obľúbený vo Veľkej Británii. Práve preto sa objavuje v seriáli Lietajúci cirkus Montyho Pythona, kde všetky položky jedálneho lístka obsahujú slovo spam a objednávky ruší skupina Vikingov spievajúcich "Spam, spam, spam..."[12, 14].

Ako spam sa označovali najprv správy mnohonásobného odosielania rovnakej správy v Usenet, neskôr sa význam začal používať pri zneužívaní správ k šíreniu rôznych textov a hlavne reklamy. Za vôbec prvý spam sa dá považovať správa rozoslaná Garym Thuerkom, marketerom firmy Digital Equipment Corporation. Správu s pozvánkou na seminár o mikropočítači DECSYSTEM-20 odoslal na 400 z 2600 adries ARPAnetu [13]. Úsmevne vyznie fakt, že niektorí vtedajší používatelia boli urazení, že spam nedostali.

5.2 Súčasnosť

Od čias Usenetu sa spam rozmohol do neskutočných rozmerov. Je to na jednej strane hrozba pre používateľa a na strane druhej veľké zaťaženie poštového servera. V súčasnosti sa používa viacero vstavaných algoritmov a filtrov, ktoré spam celkom spoľahlivo a presne identifikujú. Ale spameri hľadajú stále nové spôsoby, ako obísť všetky aktuálne spôsoby ochrany proti spamu. V dnešnej dobe je možné dokonca zakúpenie takzvanej kampane, kedy sa požadovaná správa rozošle na veľký počet adries "predávajúceho". Rádovo ide o niekoľko miliónov adries.

5.3 E-mailový spam

Je často označovaný ako junk mail a je jednou z najrozšírenejších foriem spamu. Výskyt tohto typu spamu je z celkových spamov asi 80%. Autor spamu získa adresy najčastejšie automatickým prehliadaním stránok, prípadne náhodným generátorom mailových adries. E-mailový spam môžeme rozdeliť na:

- nevyžiadaný hromadný mail (UBE),
- nevyžiadaný reklamný mail (UCE).

5.3.1 Obchodné ponuky s odkazom na WEB stránku

Je to jedna z najbežnejších foriem spamu. Ide o ponuky rôznych produktov, najčastejšie však farmaceutických produktov. Nezaostávajú ani iné "neodolateľné ponuky" na kúpu replík, online kasína či reklamy pornografických stránok alebo poistenia.

5.3.2 SCAM - Nigérijské listy

Podľa nigérijského trestného zákonníka sa označujú ako Scam 419, teda "Podvod 419". Základy tohto typu spamu položili v Nigérii okolo roku 1980, kedy bola nigérijská ekonomika zmietaná v kríze a postavená na exporte ropy. Skupinka nezamestnaných študentov tento podvod použila na manipuláciu obchodníkov v súvislosti s obchodmi s ropou. Až neskôr sa prostredníctvom e-mailu scam začal orientovať na širšiu populáciu.

Predmetom scamu je obchodná ponuka na získanie veľkého finančného obnosu ako podielu z istej obchodnej trasakcie alebo oznam o rozprávkovej výhre [16]. Podvodník sa zvyčajne z nejakého dôvodu nemôže dostať k veľkému obnosu peňazí a zaplatí vysokú províziu tomu, kto mu pomôže. Ide vždy o priamy pokus o podvod.

Zarážajúci je však fakt, že na podobné správy reaguje až 1% oslovených ľudí. Zdalo by sa, že to nie je veľa, ale pri obrovskom počte nevyžiadanej pošty je to veľké množstvo. Ak adresát reaguje na mail, podvodník od neho obratom požaduje osobné informácie a zaslanie zálohy na účet. Kópie dokladov a prípadné fotky údajného odosielateľa sú falošné alebo stiahnuté z internetu. Scameri obdržané osobné údaje môžu použiť na vytvorenie platobného príkazu pre banku a tak vybieliť účet. Ak takýto prípad nastane, tak aj napriek vypátraniu pôvodcu, poškodený svoje peniaze už nedostane pre nedostačnú legislatívu [15].

5.3.3 Kontaktné e-maily

Cieľom tejto formy spamu je vytvorenie overenej databázy e-mailov alebo následná komunikácia v duchu nigérijských listov.

5.3.4 Verifikačné e-maily

Tak isto ako u kontaktných e-mailov ide o overenie platnosti e-mailovej adresy. Takáto správa v sebe nesie zakódovanú informáciu, po kliknutí na ponúknutý hypertextový odkaz alebo stiahnutí obrázkov sa táto adresa označí ako aktívna. Je to spôsob, ako spameri zefektívňujú rozosielanie spamu a aktualizujú svoju databázu.

Najčastejším zdrojom získania e-mailových adries sú rôzne diskusné skupiny a rôzne fóra alebo webové stránky. Tieto lokality prehľadávajú roboty na to stavané. ďalším spôsobom, ako naplniť e-mailovú databázu, sú rôzne formuláre, ankety, ponuky výhier, kedy používateľ vyplní formulár za účelom získania informácie.

Za všetkým sú samozrejme peniaze. Špecializované skupiny alebo jednotlivci zbierajú e-mailové adresy za účelom predaja celej databázy alebo poskytovania služieb, tzv. kampaní na rozosielanie spamu.

5.3.5 Phishing

Názov je odvodený z anglického Password fishing - teda rybárčenie hesiel.

Už samotný názov naznačuje, čo je cieľom týchto podvodných správ, ide najmä o získanie osobných údajov, hlavne prihlasovacích údajov adresáta. Je to veľmi nebezpečná forma spamu.

Útočníci - phisheri, ako sa nazývajú, vytvárajú podvodné web stránky finančných inštitúcií, ktoré sú na prvý pohľad vernou kópiou originálnej stránky a pre bežného človeka nerozoznateľné. Jediný rozdiel je, pochopiteľne, v url adrese útočníka. Následne phisher rozošle spam s informáciou, že je treba overiť prístupové informácie, samozrejme s linkom na svoju web stránku. Adresát po zadaní svojho prihlasovacieho mena a hesla vlastne poskytne tieto údaje útočníkovi. Tieto informácie phisher zneužije vo svoj prospech [17].

Phisher môže tak isto použiť aj niektorú zo špecifickej formy malware, tzv. spyware. Infikovaný počítač potom prostredníctvom internetu posiela aktivitu používateľa. Pre útočníka sú zaujímavé hlavne prihlasovacie údaje, ale zároveň sa dostane aj k súborom uloženým na lokálnom disku, prípadne na lokálnej sieti.

5.3.6 Malware

Malware, z anglického Malicious sofware, označuje škodlivý kód vo všeobecnosti. Týmto škodlivým kódom môže byť počítačový vírus, červ, trójsky kôň, spyware a adware. Malware sa do počítača adresáta dostane najčastejšie dvomi spôsobmi:

- prijatý mail obsahuje tzv. downloader a ihneď po otvorení správy sa automaticky stiahne,
- adresát si ho stiahne vlastnou aktivitou, napríklad klikne na url adresu zaslanú v správe.

5.3.7 Nekomerčný spam

Ide o samostatnú skupinu spamu. Zaraďujeme sem správy s náboženskou a politickou tematikou, ale aj poplašné alebo reťazové správy nazývané aj hoax. Pre adresáta nepredstavujú skoro žiadne nebezpečenstvo.

5.4 Spam v mobilných telefónoch

S postupným vývinom mobilných technológií sa spam dostal aj do tejto formy komunikácie v podobe SMS. Je oveľa viac otravnejší, ale nie je tak rozšírený ako e-mailový spam.

Väčšina mobilného spamu je však pornografického charakteru, napríklad: "Miláčik, včera sme spolu hovorili", a preto sa väčšina používateľov bojí o svoj súkromný život.

S nástupom tohto druhu spamu ide ruka v ruke tak isto aj vývoj algoritmov a spôsobov, ktoré zabránia šíreniu tohto druhu spamu. To však stojí finančné prostriedky, a preto je jednoduchším riešením blokovanie čísel. Blokovanie telefónnych čísel môže ale znamenať problém v mobilnej komunikácii.

5.5 Diskusný spam

Cieľom diskusného spamu je vloženie reklamného, obťažujúceho, nežiadúceho či nerelevantného príspevku do voľne dostupných diskusných fór na internete.

Používateľ môže na väčšine blogov a spravodajských stránkach voľne napísať svoj názor, prípadne otázku autorovi. Bohužiaľ, niektorí používatelia túto skutočnosť zneužívajú a umiestňujú do diskusií príspevky, ktoré tam nepatria. Táto situácia sa ešte zhoršila s nástupom robotov. Tento spam zhoršuje komunikáciu medzi autorom článku a čitateľmi.

5.6 Spam na sociálnych sieťach a v online komunikátoroch

5.6.1 Spam na sociálnych sieťach

S nástupom sociálnych sietí na scénu dostal spam nový rozmer. Sociálne siete sú obrovským zdrojom osobných údajov a fotografií, a taktiež lákadlom pre spamerov a útočníkov.

Najjednoduchším a najbežnejším spôsobom, ako sa dostať k prihlasovacím údajom používateľa, je nasmerovanie ho na podvodnú prihlasovaciu stránku, a tak používateľ nevedomky poskytne údaje útočníkovi [18].

5.6.2 Online komunikátory

Online komunikátormi rozumieme ICQ, Skype, Windows live messenger, XMBB, Yahoo! messenger atď. Spam sa v tejto forme komunikácie niekedy zvykne nazývať aj spim.

Online komunikátory podporujú priamu slobodnú komunikáciu medzi používateľmi. Ak sa spameri dostanú k databáze používateľov niektorého z online komunikátorov, dokážu hromadne rozosielať správy s odkazmi na rôzne stránky, vírusy alebo sa snažia vylákať ďalšie informácie [19].

Takmer všetky komunikátory však podporujú tvorbu white listov, a tak zamedzia tejto forme spamu.

6 Spôsoby detekcie spamu

Detekcia spamu sa vyvíjala spolu s formami rozosielania spamu. Spôsoby detekcie môžeme rozdeliť na 2 spôsoby:

- identifikovať spam podľa odosielateľa, či to bude už jeho IP adresa, alebo dôveryhodnosť servera odosielateľa,
- identifikovať spam podľa tela správy.

Na detekciu sa najčastejšie používajú kombinácie automatických filtrov. Úspešnosť metódy je udávaná v percentách úspešnej identifikácie spamu - false positive. Veľmi dôležitým faktorom je však aj označenie normálnej pošty ako spam - false negative. Označenie spamu za normálnu poštu nie je v menšom množstve taký veľký problém ako označenie normálnej pošty za spam, pretože menší počet spamov si užívateľ vie ľahko filtrovať aj sám, ale ak sa pošta dostane do spamu, užívateľ si ju zväčša nevšimne vôbec.

6.1 Pravidlový systém - rule base

Je jedným z najstarších spôsobov detekcie spamu. Na detekciu spamu sa používali hlavne regex výrazy. Vytvárali sa akési súbory pravidiel, na základe ktorých bol spam identifikovaný.

V začiatkoch bola táto metóda veľmi účinná, no s nástupom sofistikovanejších metód rozosielania hromadných správ, ako napríklad vkladanie náhodných slov či textových reťazcov do textu, stratila účinnosť. Dnes pri použití samotnej metódy slovníku je úspešnosť detekcie asi 30%.

V dnešnej dobe je vytváranie pravidiel časovo náročné a neefektívne, napriek tomu nájdeme zástupcu tohto zastaralého systému, a tým je opensource projekt The Apache SpamAssassin Project. Vytvorené pravidlá nebolo možné tak dynamicky meniť a prispôsobovať týmto praktikám. Táto metóda patrí k metódam detekcie na základe tela správy [20].

6.2 Systém založený na odtlačkoch - fingerprint

Pomocou matematickej funkcie bol vypočítaný napríklad hash a na základe toho sa spam detekoval. Firmy sa špecializovali na to, že spam zachytávali a prichádzajúce správy porovnávali s databázou, ktorú mali k dispozícii.

Problémy tejto metódy začali, keď sa začali používať novšie metódy rozosielania hromadnej nevyžiadanej pošty a začali sa do spamov, ktoré už v databáze existovali, vkladať náhodné textové reťazce, čím sa zmenil kontrolný súčet a správa bola označená ako žiadúca. Na toto, samozrejme, reagovali aj firmy, ktoré začali robiť mäkšie odtlačky spamu, čím sa ale rapídne zvýšil počet false negative detekcií, teda označenie normálnej pošty za spam. Táto metóda tak analyzuje telo správy [21].

6.3 Čierne zoznamy - black lists

Ide o metódu detekcie založenú na zbieraní IP adries, z ktorých je spam odosielaný. Jedná sa o zbieranie IP adries a ich uchovanie. Rozšírením tohoto spôsobu je zdieľanie zoznamov pomocou DNS protokolu. Tieto služby sú voľne dostupné na internete. Problémom býva fakt, že spam je odosielaný z falošných IP adries, a tak môže byť v konečnom dôsledku poškodený nevinný. Úspešnosť black listov je približne 10% [22].

S touto metódou je úzko spojená tvorba tzv. white listov, teda zoznam dôveryhodných IP adries. Je to presný opak black listov. Úspešnosť white listov je 100%. White listy sú však vhodné iba pre koncového užívateľa [25].

6.4 Systém založený na klasifikácii odosielateľa - reputation service

Táto metóda je založená na tom, že adresné priestory providerov sa navzájom poznajú. Môže ísť tak isto o rezervované adresné priestory, teda priestory, z ktorých by mail prísť nemal. Tak isto by nemal prísť z adresného priestoru, ktorý je určený priamo užívateľovi. Na základe tejto klasifikácie adresných priestorov sa systém môže rozhodnúť, či bude mail označený ako spam, alebo nie [23].

6.5 Šedý zoznam - grey list

Nejde priamo o zoznam ako tomu bolo pri black listoch. Systém dokáže rozlíšiť, či ide o direct mail, teda mail odosielaný priamo užívateľom, alebo použitie open relay serveru. Na rozlíšenie spam generátora od korektného mailserveru sa používa zavedenie chyby do komunikácie, kedy sa očakáva, že táto chyba bude akýmsi spôsobom ošetrená druhou stranou. Klasický mailserver sa znovu pokúsi komunikáciu nadviazať, ale spam generátor zvyčajne nie. Spam roboty sú stavané na to, aby odoslali čo najväčšie množstvo spamu v čo najkratšom čase a zvyčajne neošetrujú všetko, čo by mali pri komunikácii ošetriť. Grey list dosahuje úspešnosť nad 95%. Nevýhodou tohto filtra je zvýšený čas doručenia o niekoľko desiatok minút až hodín [24].

6.6 Štatistické systémy - Baysov filter

Táto metóda je založená na tom, že tieto systémy počítajú na základe akejsi štatistiky, či je daný mail spam, alebo nie je spam. Detekciu uskutočnujú na základe výskytu slov alebo skôr kombinácií slov v maili. Metóda je agilná pri zmenách poradia slov či pri vkladaní náhodných slov alebo textov do správy. Táto metóda nepozerá na IP adresu odosielateľa. Štatistické metódy majú veľkú úspešnosť detekcie a ľahko sa prispôsobia a prekonvertujú celú databázu na nové pravidlá pri určovaní spamu [27]. V závislosti od implementácie majú úspešnosť okolo 99%.

6.7 Novodobé spôsoby - SPF, DKIM

6.7.1 SPF - server policy framework

SPF - server policy framework. Nie je to priamo metóda, ktorá filtruje spam, ale umožňuje detekovať falošne zadaný e-mail odosielateľa. Užívateľ si môže sám nastaviť rozsah adresných priestorov, z ktorých bude odosielať poštu, čo je najväčšou výhodou tohto systému. Nevýhodou môže byť to, že ho používateľ môže aj deaktivovať, čím sa stane zbytočný. Medzi nevýhody radíme aj to, že ak sa útočník dostane na server, kde hostuje veľa domén a každej tento systém deaktivuje - nastaví neobmedzený rozsah adries, z ktorých môžu e-maily prijímať, môže na všetky tieto adresy rozosielať spam neobmedzene [28].

6.7.2 DKIM - DomainKeys Identified Mail

DKIM - DomainKeys Identified Mail. Tento spôsob ochrany proti spamu je stavaný na dvojici kľúčov - verejných a súkromných. Súkromným kľúčom sa mail podpisuje na SMTP serveri, či už firemnom, alebo u poskytovateľa a nie je potrebná žiadna súčinnosť zo strany príjemcu. Podpísaný e-mail v sebe nesie informáciu, kde sa podpis nachádza, a tak isto, kde si môžeme tento podpis stiahnuť - verejný kľúč. Je to zároveň aj nevýhoda systému, pretože spamerovi nič nebráni tomu, aby si tento e-mail podpísal sám, a tak isto na svoje servery umiestnil korektný podpis. Druhou slabou stránkou tejto metódy je fakt, že či je e-mail podpísaný, sa dozvieme až z tohoto e-mailu [29].

6.8 Neurónové siete

Najmladším spôsobom detekcie spamu sú takzvané neurónové siete. Nevýhodou neurónových sietí je vysoká náročnosť na výkon a práve táto nevýhoda odrádza od implementácie tohto systému. Táto metóda využíva umelú inteligenciu, dokáže sa ”naučiť” pravidlá na zistenie nevyžiadanej pošty [30].

7 Použitie n-gramov na detekciu spamu

Detekovať spam pomocou extrakcie n-gramov je možné, ale skôr má zmysel, pokiaľ vynecháme prílohy a obrázky. Spracovávame teda len textovú časť spamov. Ide o problém porovnávania a spracovávania textov. Oproti porovnávaniu textov máme jednu veľkú výhodu, a to v podobe dĺžky e-mailových správ. Tie sú podstatne kratšie ako textové dokumenty, a preto je aj práca s nimi menej náročná na výkon. Postup algoritmu môžeme vidieť na obrázku č. 3

Našou úlohou je nájsť spoločné časti porovnávanej správy a jednotlivých správ v databáze. Na základe tejto podobnosti dokážeme zistiť, či je daná správa spamom, alebo do akej miery. Samotné rozhodnutie necháme však na používateľa. Používateľ sa rozhodne, či vložime správu do databázy, alebo nie.

7.1 Predspracovanie textu

Predspracovaním textu rozumieme predspracovanie textového dokumentu. Výsledkom analýzy je text spracovaný na požadovaný výstup. Je to prvá, a zároveň veľmi dôležitá úloha pri spracovaní textov.

Ak chceme detekovať spam na základe podobnosti n-gramov, je táto časť algoritmu tak isto jedna z najdôležitejších. Pri spame sa často jedná o strojový preklad a z toho dôvodu je potrebné:

- odstránenie diakritiky,
- odstránenie veľkých písmen,
- odstránenie nezmyselných znakov.

Odstránenie diakritiky bude prvoradou úlohou, hlavne pri analýze slovenského a českého spamu.

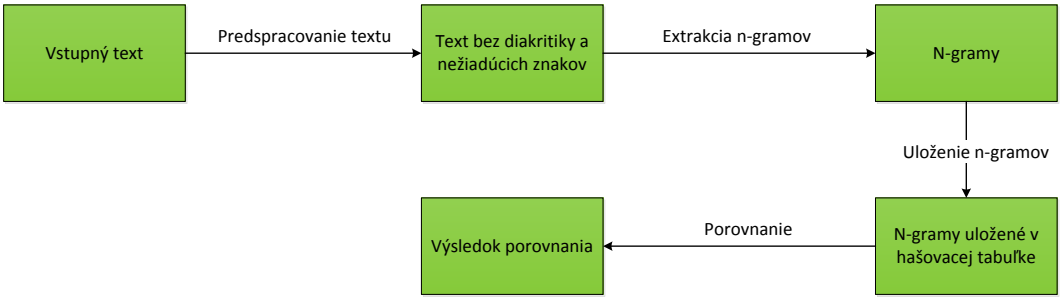
Následne máme pred extrakciou n-gramov 2 možnosti:

- Celú databázu rozdelíme iba na jednotlivé správy - výhodou je zachytenie vetných presahov. Nevýhodou je zvýšený počet n-gramov pri následnej extrakcii.
- Jednotlivé správy rozdelíme ešte na vety - výhodou je menší počet extrahovaných n-gramov. Nevýhodou je nezachytenie súvislostí medzi vetami.

V našej práci otestujeme obe možnosti a na záver zhrnieme výsledky aj v porovnaní s inými nástrojmi na detekciu spamu.

7.2 Výber vhodnej dĺžky n-gramov

Ak máme text už pripravený na extrakciu n-gramov, je treba nájsť a zvoliť vhodnú dĺžku n-gramov. Treba však brať do úvahy viacero faktorov:



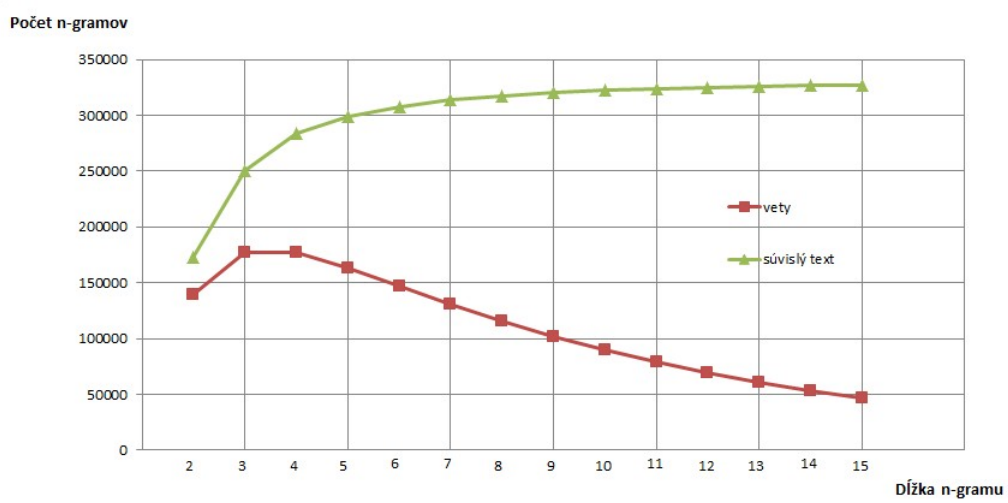
Obrázek 3: Postup algoritmu

- Kratšie n-gramy - napríklad bigramy alebo trigramy môžu zachytávať bežné slovné spojenia a krátke frázy. Extrahovaním n-gramov tejto dĺžky dostaneme veľkú databázu, v ktorej nájdeme veľa opakujúcich sa n-gramov. Rovnaké kratšie n-gramy môžu obsahovať viaceré správy a takéto porovnanie je potom nepresné.
- Dlhšie n-gramy - 8-gramy, 9-gramy či 10-gramy zachytávajú v e-mailovej komunikácii častokrát celé vety, prípadne aj vetné presahy. Extrahovaním dostávame veľmi malú vzorku n-gramov. Táto vzorka n-gramov je veľmi malá a pri krátkych správach veľmi nepresná.
- Ideálna dĺžka n-gramov - 4-gramy, 5-gramy, 6-gramy, 7-gramy. Kde 4-gramy predstavujú pomyselnú dolnú hranicu a 7-gramy zasa vrchnú hranicu ideálnej dĺžky n-gramov pre toto použitie.

V našej práci sme sa zamerali práve na 4-gramy a 7-gramy. Podobne ako vo výskumoch [41], bude však možné použiť akúkoľvek dĺžku.

Na obrázku č. 3 vidíme, ako sa vyvíja počet extrahovaných n-gramov z databázy, ktorú máme k dispozícii. Vývoj počtu n-gramov po rozdelení jednotlivých správ na vety sa s nárastom veľkosti n-gramu znižuje z výnimkou pri veľkosti n-gramu 2. Tento vývoj je daný dĺžkou viet v správach. Ak je veta dlhšia ako dĺžka n-gramu, program ju jednoducho preskočí a pokračuje ďalšou vetou. Týmto sa postupne vylučujú krátke vety a tým pádom sa rapídne znižuje aj počet n-gramov, ktoré dostaneme extrakciou z celej databázy. Tak isto to odzrkadluje už spomínaný fakt, že vety v e-mailových správach sú podstatne kratšie ako napríklad v literárnych dielach.

Pri rozdelení celej databázy na správy, teda zachovanie súvislého textu správ, je situácia iná. Pri bigramoch je počet extrahovaných n-gramov len o čosi vyšší ako pri rozdelení na vety. Rozdiel je tvorený zachytením vetných presahov. Výrazné stúpanie vidíme iba po 5-gramy následne je stúpanie počtu n-gramov len mierne.



Obrázek 4: Vývoj počtu n-gramov

7.3 Extrakcia n-gramov

Veľmi dôležitým krokom je voľba správneho algoritmu na extrakciu n-gramov. Je dôležité zachytiť všetky možné n-gramy, aby bola táto metóda odolná voči posunom textu, vkladaniu nových slov či prípadnú zámenu slov.

Rozdelenie textu na n-gramy rôznej dĺžky sa pokúsime priblížiť na nasledujúcom príklade. Máme vetu: "Kryptomena bitcoin sa teší stále väčšej popularite aj napriek nestabilnému kurzu." Po rozdelení tejto vety na n-gramy s dĺžkou 3 slová dostaneme nasledovné 3-gramy: "Kryptomena bitcoin sa — bitcoin sa teší — sa teší stále — teší stále väčšej — stále väčšej popularite — väčšej popularite aj — popularite aj napriek — aj napriek nestabilnému — napriek nestabilnému kurzu". Pomocou tejto metódy teda dostaneme všetky možné n-tice slov z textu. Prípadnou zmenou jedného slova v texte dostaneme práve n odlišných n-gramov. Ak do vety slovo doplníme dostaneme práve n nových n-gramov, ktoré určite nebudú v databáze. Pri prehodení slovosledu vo vete je toto číslo menšie a bude závisieť od dĺžky vety a n-gramu.

7.4 Uloženie n-gramov

Nemenej dôležitým krokom je výber vhodnej štruktúry na uloženie získaných n-gramov. V tomto prípade je vhodnou štruktúrou práve tá, ktorá nám umožní rýchle nájdenie zhodného n-gramu pri meraní.

Na uloženie extrahovaných n-gramov sme použili vstavanú triedu .NET frameworku Hashtable. Má všetky predpoklady na presné a rýchle porovnanie zhodných n-gramov, pretože jej primárnou funkciou je práve vyhľadávanie hodnôt na základe kľúča [4]. Pre hašovaciu tabuľku sme sa rozhodli na základe kritérií:

- Jednoduchá implementácia - jednoduchšia ako pri b - stromoch alebo iných dátových štruktúrach.
- Rýchle vkladanie - dáta sa neudržiavajú usporiadané na rozdiel od iných údajových štruktúr.
- Rýchle vyhľadávanie - priemerná časová zložitosť O, zriedkavá najhoršia zložitosť $O(n)$

Kľúčom bude samotný n-gram a hodnotou bude kolekcia správ, v ktorých sa daný n-gram nachádza. Týmto krokom odstránime duplicity v jednej správe, a zároveň dosiahneme to, že jeden n-gram môže obsahovať viacero správ.

Kolekcia správ, ktorá je načítaná pri spustení programu je doplnená o poradové číslo správy. Toto číslo môže byť pri každom spustení unikátne. V hašovacej tabuľke sú ako hodnoty ukladané práve čísla správ prislúchajúce danému n-gramu. Pri ukladaní merania naopak uložíme celú správu bez poradového čísla. Túto správu nájdeme jednoduchým LINQ výrazom na základe jej indexu v kolekcií načítanej pri spustení.

7.5 Porovnanie n-gramov

Pre samotné porovnanie n-gramov je dôležité nielen extrahovať n-gramy z celej databázy, ale aj rovnakým spôsobom získať n-gramy z textu, ktorý chceme porovnávať.

Ak máme n-gramy z porovnávaného textu, jednoduchým spôsobom ich porovnáme s databázou. Výsledkom porovnania bude kolekcia indexov jednotlivých správ, ktoré obsahujú. Celkový počet zhodných n-gramov bude súčet všetkých indexov. Pre naše účely bude však hodnotnejší údaj zhoda s jednotlivými správami. Túto zhodu nájdeme na základe početnosti.

7.6 Váhovanie

Váhovaním rozumieme rozhodovanie, do akej miery je porovnávaný text zhodný v našom prípade, či je šanca, aby bola daná správa spamom. Hneď na začiatku si však musíme uvedomiť, že detekcia bude závislá na databáze. Budeme vedieť detekovať len spamy, ktoré sú aspoň minimálne zhodné s niektorou zo správ v databáze.

Výsledkom merania našej aplikácie bude zobrazenie piatich najzhodnejších správ meranej správy s databázou. Zhoda bude vyjadrená ako podiel počtu zhodných n-gramov danej správy s porovnávanou správou a celkového počtu n-gramov danej správy. Doplnkovým údajom bude vyjadrenie v percentách.

Či a nakoľko je správa spamom, musíme rozhodnúť práve na základe týchto údajov. Keďže je dĺžka n-gramov variabilná, musíme správne odhadnúť aj skutočnosť, že pri väčšej dĺžke n-gramov sa zníži počet zhodných n-gramov so správou aj počet n-gramov, ktoré dostaneme zo správy. Preto je dôležité kritérium na označenie podozrivej správy zdvojiť. Prvým kritériom bude práve počet n-gramov zhodných s porovnávanou správou a druhým bude percentuálne vyjadrenie.

V našom programe nastavíme váhy detekcie nasledovným spôsobom. Najdôležitejšie je porovnanie s najzhodnejšou správou:

- Správa je spamom, ak je zhoda 50% a viac alebo počet zhodných n-gramov je 10 a viac,
- správa je podozrivá, ak je zhoda v rozmedzí 35% - 50% alebo počet zhodných n-gramov je v rozmedzí 7 - 10,
- správa je minimálne zhodná, ak je zhoda v rozmedzí 10% - 35% alebo počet zhodných n-gramov je v rozmedzí 4 - 7,
- spáva nie je spamom v prípade, ak je zhoda menšia ako 10% alebo počet zhodných n-gramov je menší ako 4.

Nastavenie dvojitých váh je výhodné z toho dôvodu, že e-mailové správy sú rôznej dĺžky. Percentuálne vyjadrenie zhody má význam pri kratších správach, kedy spam nemusí dosiahnuť dostatočný počet n-gramov na detekciu, ale pritom je z veľkej časti zhodný so správou v databáze. Naopak, pri správach, ktoré sú dlhšie, prípadne sa skladajú z viacerých správ, je výhodnejšie hľadať zhodu na základe počtu n-gramov. Veľkú úlohu zohráva aj samotná zvolená dĺžka n-gramov.

7.7 Použité jazyky

Algoritmus je navrhnutý tak, aby sme mohli do databázy vložiť správu v akomkoľvek jazyku. Musíme však brať do úvahy aj už vyššie spomínanú skutočnosť, že pokiaľ nebudeme mať dostatočný korpus zo správ daného jazyka, meranie bude veľmi nepresné. Primárne sa ale predpokladá, že správy budú hlavne v troch jazykoch a tými sú: čeština, slovenčina a angličtina.

7.8 Zdroj dát

Základom databázy spamov sú zozbierané spamy z vlastných e-mailových adries, spamovej zložky e-mailovej adresy futbalového klubu MKT Leopoldov, e-mailových adries Hotela Máj, najväčšia časť spamových správ je však z portálu hoax.cz. Z portálu hoax.cz boli použité sekcie phishing, scam, hoax a lotérie.

Pre moju prácu som zozbieral celkom 1685 správ. Správy sú v rôznych jazykoch. Najčastejšie sú však v slovenčine, češtine a angličtine. V databáze sa nachádzajú aj správy v iných jazykoch, ako napríklad v nemčine, francúzštine, ruštine či maďarčine. Databáza je uložená v súbore XML. XML má nasledujúcu šablónu:

```
<message>
   <sender> odosielatel </sender>
   <recipient> prijemca </recipient>
   <subject> predmet </subject>
   <body> telo spravy </body>
</message>
```

Výpis 1: Ukážka uloženia správ vo formáte XML

Týmto spôsobom sú ukladané všetky správy v databáze. Pri spustení programu sa celá databáza načíta do kolekcie, pri načítaní sa každej správe vytvorí index, na základe ktorého bude program pracovať so správami.

Podobným spôsobom sú ukladané aj jednotlivé merania. Ako databáza je teda použitý jednoduchý súbor formátu XML. Pri ukladaní meraní je jedinečnou hodnotou dátum a čas merania, podľa ktorého sú merania zoradené.

7.9 Popis algoritmu porovnania n-gramov

V našom algoritme najskôr vo vstupnej správe odstránime diakritiku, veľké písmená a nežiadúce znaky. Odstránenie nežiadúcich znakov závisí od predvoľby merania, ak si prajeme text rozdeliť na vety, ponecháme v texte bodky, ktoré označujú koniec vety.

V nasledujúcom kroku sa text rozdelí na časti, ktoré oddeľujú bodky a jednotlivé časti uloží do poľa. Teda, pokiaľ sme zvolili možnosť zachovania súvislého textu správ, čítač uloží do poľa celý text správy, a tak zachová vetné presahy.

Ďalší krok postupne z tohoto poľa berie textové reťazce a delí ich na slová. Rozdeľovacím znakom je v tomto prípade prázdny znak. Vstupným parametrom je veľkosť n-gramu vyjadrená číslom. V našom prípade to znamená, že algoritmus vyberá spôsobom popísaným vyššie n-tice slov a tieto n-tice postupne vkladá do hašovacej tabuľky, pričom kľúčom je samotný n-gram tvorený n-ticou slov a hodnotou je kolekcia správ, v ktorých sa nachádza daný n-gram. Takto sa vytvára hašovacia tabuľka pri extrakcii n-gramov z databázy. Pri vkladanej správe sa vytvorí iba pole n-tíc požadovanej veľkosti.

Pri porovnávaní n-gramov získaných z vloženej správy a uložených v poli porovnávame v cykle každý n-gram s pripravenou hašovacou tabuľkou nasledovným spôsobom:

- Ak sa n-gram v tabuľke nenachádza, prejdeme na ďalší,
- ak sa n-gram v tabuľke nachádza, tak zistíme, v akých správach sa nachádza na základe hodnoty kľúča a uložíme ho do kolekcie.

Výsledkom tohoto kroku je kolekcia indexov správ. Jednoduchým LINQ výrazom si túto kolekciu roztriedime a nájdeme 5 najzhodnejších správ, teda po správnosti ich indexov. Pri ukladaní merania si podľa indexov nájdeme zhodnú správu a túto správu bez indexu uložíme. Príslušný LINQ výraz vidíme na nasledujúcom výpise:

```
var pocet = (from cisloEmail in zhody group cisloEmail by cisloEmail into g
    select new {EmailCislo = g.Key, Pocet = g.Count()}).OrderByDescending(x => x.Pocet).Take(5);
```

Výpis 2: LINQ výraz na výber 5 zhôd

8 Návrh aplikácie

Na písanie našej aplikácie sme zvolili programovací jazyk c# a vývojové prostrenie Visual Studio 2012. Ide o klasického desktopového klienta využívajúceho .NET framework vo verzii 4.5. Dáta potrebné na chod aplikácie sú ukladané vo formáte XML. Aplikácia dokáže pracovať offline, ale pre načítenie spamovej zložky z e-mailovej adresy je, samozrejme, potrebné internetové pripojenie.

8.1 Hlavný formulár

Ihneď po spustení aplikácie sa načíta celá databáza správ do kolekcie, aby bola ihneď pripravená na prácu a zjaví sa základné okno aplikácie, ktoré môžeme vidieť na obrázku č. 5.

Horné menu obsahuje položky Správa databázy a Detekcia. V správe databázy máme na výber položky, ktoré budú spomenuté nižšie a tými položkami sú Pridanie spamu samostatne, Pridanie spamu načítaním z e-mailovej adresy a Štatistika.

Na ľavej strane sa nachádzajú možnosti merania Rozdelenie a Dĺžka n-gramu. Pri rozdelení máme na výber 2 možnosti:

- rozdelenie správ na vety,
- zachovanie súvislého textu správ.

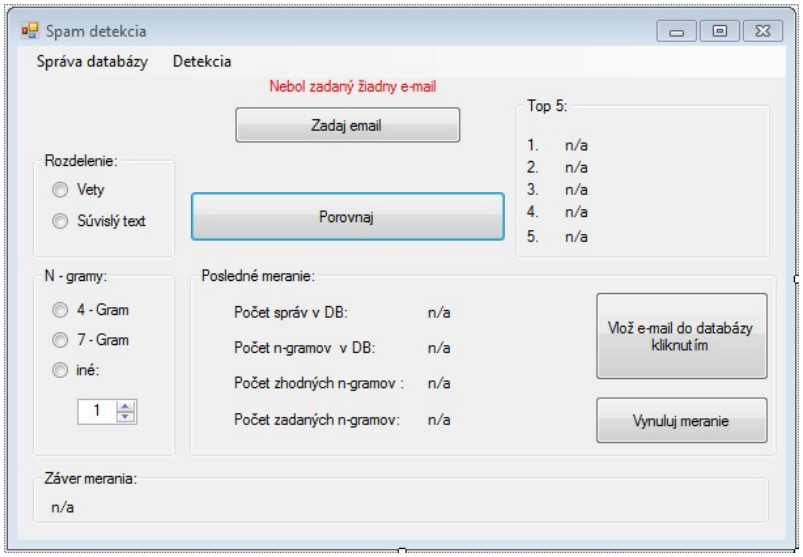
Dĺžka n-gramu má na výber 3 možnosti:

- extrakcia 4-gramov,
- extrakcia 7-gramov,
- vlastná dĺžka n-gramov v rozmedzí 2-20.

V strednej časti úplne navrchu sa nachádzajú 2 tlačidlá a chybový indikátor, ktorý nám oznámi chybový stav, pokiaľ chceme porovnať prázdny e-mail s databázou. Tlačidlo Zadaj mail nám otvorí formulár na zadanie mailovej správy, ktorú chceme porovnať. Po tom, ako zadáme správu na porovnanie, môžeme správu porovnať s databázou stlačením tlačidla Porovnať.

Po úspešnom vykonaní merania sa posledné meranie zobrazí v hlavnom formulári. Konkrétne na ľavej strane v sekciách Posledné meranie a Top 5. V sekcii Posledné meranie sa tak isto sprístupnia 2 tlačidlá na vynulovanie tohoto merania a vloženie porovnávanej správy do databázy. Sekcia Top 5 nám zobrazí najväčšie zhody porovnávanej správy s databázou. Zobrazí pomer počtu zhodných n-gramov a všetkých n-gramov správy v databáze vyjadrené aj v percentách. Tieto zhody sú zoradené podľa počtu zhodných n-gramov, nie podľa percentuálnej zhody.

Spodná časť formulára je vyhradená pre záver merania. Zobrazí odporúčaný záver merania.



Obrázek 5: Hlavné okno aplikácie

8.2 Formulár pre zadanie e-mailu

Na obrázku č. 6 vidíme formulár na vkladanie správy. Vkladanie správy na porovnanie a vkladanie správy do databázy spamu používa rovnaký formulár.

Vkladaná správa musí obsahovať adresu odosielateľa, adresu príjemcu a samotné telo správy. Predmet môže zostať prázdny.

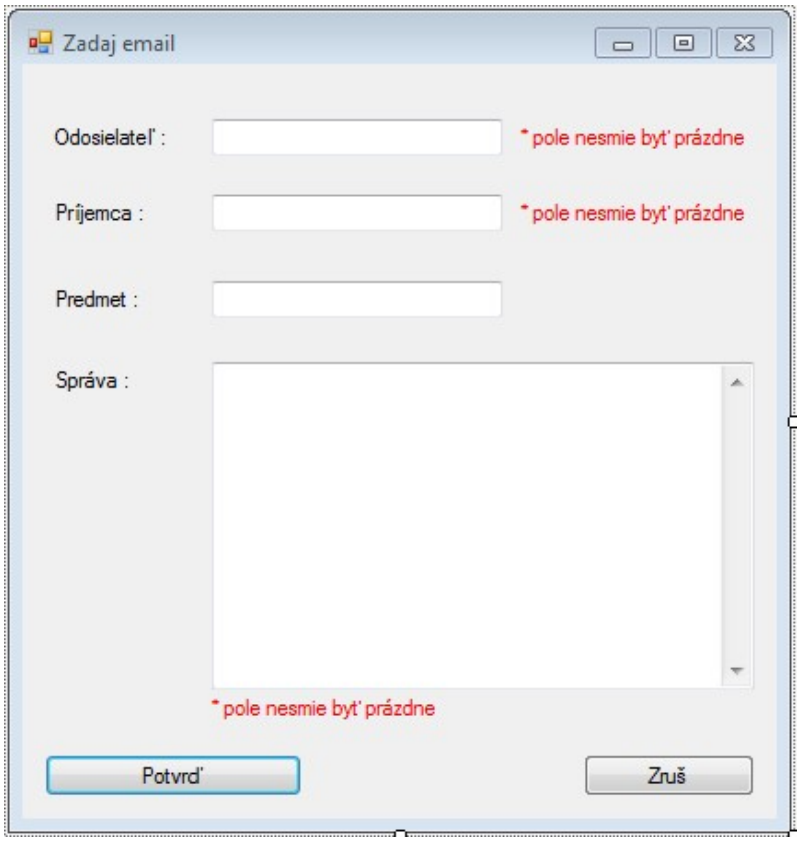
8.3 Načítanie spamu z mailovej adresy

Pri načítaní spamov z mailovej adresy sme použili knižnicu Mail.dll. Kompletnú dokumentáciu a popis knižnice nájdeme na webovej stránke autorov [40].

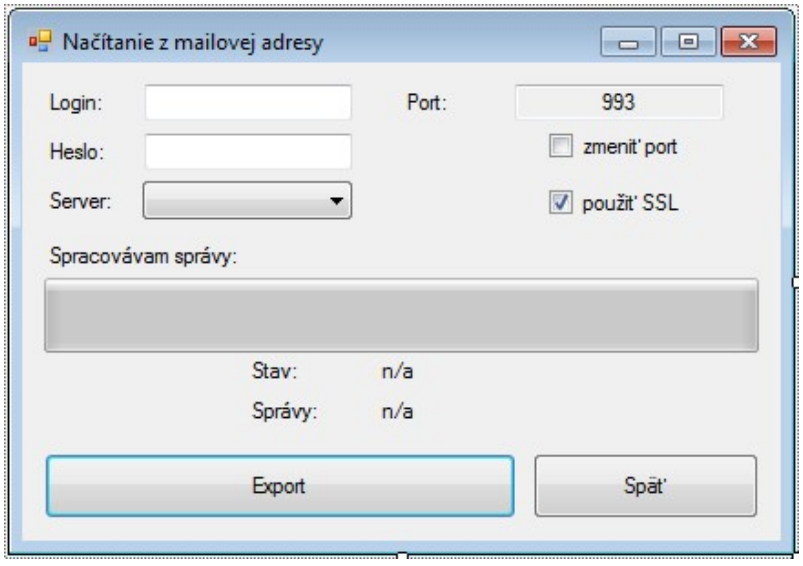
Načítanie správ je možné z dvoch najpoužívanejších voľne dostupných mailových schránok, ktorými sú český seznam.cz a americký googlemail.com. V oboch prípadoch sa v základe používa zabezpečený protokol SSL a port 993. Nastavenie je však možné zmeniť. Priebeh spracovávania správ vidíme ako progres status baru. Nakoniec je používateľ oboznámený o výsledku. Náhľad na tento formulár vidíme na obrázku č. 7

8.4 Formulár meraní

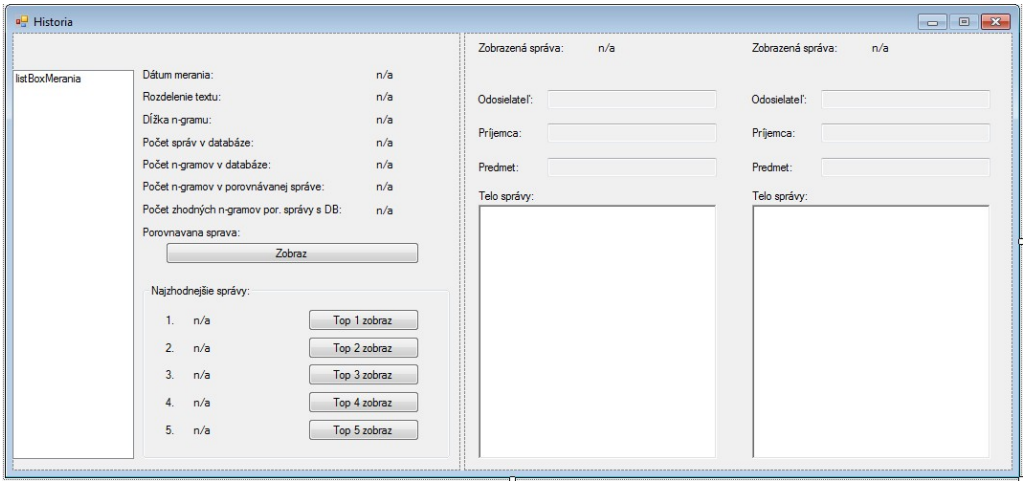
Formulár meraní obsahuje kompletné informácie o všetkých vykonaných meraniach. Merania sú zoradené podľa dátumu vykonania. Všetky sa nachádzajú na ľavej strane.



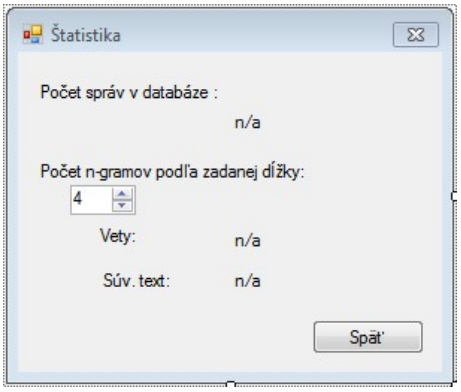
Obrázek 6: Formulár pre zadanie e-mailu



Obrázek 7: Formulár pre načítanie spamov z mailovej adresy



Obrázek 8: História meraní



Obrázek 9: Štatistika

8.5 Štatistika

Najjednoduchším formulárom programu je štatistika databázy, ktorá obsahuje aktuálny počet správ nachádzajúcich sa v databáze a počty n-gramov na základe zadanej dĺžky a rozdelenia, prípadne nerozdelenia textu na vety.

9 Výsledky

Pre testovanie úspešnosti detekcie nášho algoritmu sme si vybrali 5 spamových správ a 5 správ, ktoré by program mal označiť ako normálnu poštu. Testovali sme n-gramy dĺžky 4 a 7. Tak isto sme skúšali text rozdeliť na vety a zachovať vetné presahy. Jednotlivé tabuľky obsahujú počet n-gramov porovnávanej správy, počet zhodných n-gramov s celou databázou a 5 najzhodnejších správ. Správy s najväčšou zhodou sú v databáze zobrazené v tabuľkách ako počet zhodných n-gramov/počet všetkých n-gramov podobnej správy. Výsledky meraní sú zhrnuté v nasledujúcich tabuľkách.

V tabuľkách 1-4 sme testovali detekciu na 5 spamoch. Výsledky sa líšili podľa spôsobu extrakcie n-gramov z databázy. Najefektívnejšia detekcia je pri extrahovaní 4-gramov z textu a pri zachovaní súvislého textu. Pri tejto metóde algoritmus úspešne identifikoval tri správy z piatich. Porovnávaná spamová správa mala najviac zhodných n-gramov s najpodobnejšími správami. Iba v dvoch prípadoch mali porovnávané správy zhodné menej ako 4 n-gramy s jednotlivými správami. Podobne obstála aj extrakcia 4-gramov pri rozdelení textu na vety, kedy tak isto detekovala 3 z 5 spamov. Pri tejto metóde boli výsledky podobné. Tak isto v dvoch prípadoch nemala porovnávaná správa ani 4 n-gramy zhodné so žiadnou zo správ v databáze. Pri jednej zo správ bola zhoda jednoznačná a pri zvyšných dvoch bola zhoda na hranici nastavených váh.

Zaujímavé výsledky vidíme pri rozdelení textu na dlhšie časti, konkrétne 7-gramy. Pri rozdelení textu na vety dokázal algoritmus detekovať iba 1 spam z 5. Pri ostatných nenašiel žiadnu alebo takmer žiadnu zhodu. Pri zachovaní súvislého textu je výsledok o trošku lepší a algoritmus detekuje 2 z 5 spamov. Ostatné tak isto nedetekujú žiadnu alebo len zanedbateľnú zhodu, to znamená jeden alebo dva zhodné n-gramy.

V tabuľkách 5-8 sme testovali detekciu na piatich správach, ktoré spamom nie sú. Pri týchto meraniach boli výsledky očakávané. Ani jedna zo správ nemala žiadnu zhodu s databázou. Potvrdilo sa tak to, že spamy sa zväčša podobajú a sú odlišné od normálnej pošty.

Úspešnosť nášho algoritmu založeného na n-gramoch je na základe tabuliek okolo 60%. Berieme teda do úvahy iba rozdelenie textu na 4-gramy. Úspešnosť pri 7-gramoch bola približne 20 - 40%.

9.1 Porovnanie s inými metódami detekcie

Ak porovnáme náš algoritmus s inými metódami, pri súčasnom stave databázy zistíme, že nedosahuje výsledky ako momentálne najlepšie nástroje na filtrovanie spamu. Stručné porovnanie s vybranými metódami popísanými v našej práci si môžete pozrieť v tabuľke č. 9.

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	20	10	5/20	5/20	-	-	-
2.	57	9	2/345	1/408	1/289	1/484	1/237
3.	10	9	2/185	1/85	1/461	1/385	1/205
4.	563	132	4/283	4/173	4/266	4/333	4/217
5.	41	35	33/39	2/54	-	-	-

Tabulka 1: Rozdelenie na 4-gramy a na vety

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	59	26	13/61	13/61	-	-	-
2.	84	9	2/476	1/524	1/374	1/600	1/305
3.	25	9	2/293	1/133	1/662	1/638	1/312
4.	817	137	4/675	4/443	4/639	4/706	4/688
5.	75	74	66/75	8/82	-	-	-

Tabulka 2: Rozdelenie na 4-gramy a zachovanie súvislého textu

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	6	0	-	-	-	-	-
2.	36	0	-	-	-	-	-
3.	5	0	-	-	-	-	-
4.	377	2	1/16	1/107	-	-	-
5.	21	14	14/19	-	-	-	-

Tabulka 3: Rozdelenie na 7-gramy a na vety

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	56	18	9/58	9/58	-	-	-
2.	81	0	-	-	-	-	-
3.	22	0	-	-	-	-	-
4.	814	2	1/101	1/314	-	-	-
5.	72	57	57/72	-	-	-	-

Tabulka 4: Rozdelenie na 7-gramy a zachovanie súvislého textu

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	58	0	-	-	-	-	-
2.	52	0	-	-	-	-	-
3.	57	0	-	-	-	-	-
4.	31	0	-	-	-	-	-
5.	64	38	4/250	2/293	2/396	2/350	2/323

Tabulka 5: Rozdelenie na 4-gramy a na vety

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	127	0	-	-	-	-	-
2.	87	0	-	-	-	-	-
3.	119	0	-	-	-	-	-
4.	65	0	-	-	-	-	-
5.	62	65	4/324	3/528	2/393	2/442	2/419

Tabulka 6: Rozdelenie na 4-gramy a zachovanie súvislého textu

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	28	0	-	-	-	-	-
2.	28	0	-	-	-	-	-
3.	29	0	-	-	-	-	-
4.	12	0	-	-	-	-	-
5.	25	0	-	-	-	-	-

Tabulka 7: Rozdelenie na 7-gramy a na vety

č. správy	počet n-gramov	počet zhod.	top1	top2	top3	top4	top5
1.	124	0	-	-	-	-	-
2.	84	0	-	-	-	-	-
3.	116	0	-	-	-	-	-
4.	62	0	-	-	-	-	-
5.	59	0	-	-	-	-	-

Tabulka 8: Rozdelenie na 7-gramy a zachovanie súvislého textu

metóda	úspešnosť	klady	zápory
Pravidlový systém	približne 30%	Jednoduchá tvorba pravidiel	Náročné udržiavanie aktuálnosti pravidiel
Čierne zoznamy	približne 10%	Dostupnosť zdielaných databáz na internete	Závislosť na aktuálnosti databáz
Biele zoznamy	100%	Vhodné iba pre súkromné použitie	Úzky okruh odosielateľov
Bayesov filter	približne 99%	Schopnosť učenia a prispôsobivosť	Chybovosť pri zaraďovaní správ
Šedé zoznamy	cez 95%	Šetrenie dátového prenosu	Doručovanie správ s oneskorením
SPF	vysoká	Jednoduché zavedenie	Obmedzenie na používanie oficiálnych SMTP serverov
Naša metóda	60%	Jednoduché pridávanie spamov. Takmer nulové false positive	Úzka závislosť na databáze

Tabulka 9: Porovnanie metód detekcie spamu

10 Záver

Cieľom tejto práce bolo vypracovať rešerš týkajúcu sa použitia n-gramov na spracovanie textových dokumentov, navrhnúť a implementovať aplikáciu, ktorá dokáže detekovať spam na základe metódy extrakcie n-gramov a nakoniec výsledky porovnať so súčasnými metódami detekcie.

V jednotlivých kapitolách sme vysvetlili pojem n-gram a uviedli všestranné použitie n-gramov v rôznych oblastiach, napríklad kategorizácia textov, analýza DNA, analýza hudby, rozpoznanie jazyka a i. Popísali sme dátové štruktúry ukladania n-gramov a algoritmy na extrakciu n-gramov, z ktorých najznámejšie sú Nagao 94, LZW algoritmus, sufixové pole, invertovaný index a i. ďalším dôležitým pojmom je spam a spôsoby jeho detekcie. Predstavili sme vývoj spamu od jeho vzniku prvou masovo rozoslanou správou v sieti ARPANET až po súčasnú podobu automatického zasielania správ na obrovské počty adries, najčastejšie za účelom zisku. V práci sme popísali aj spôsoby detekcie spamu používané v začiatkoch až po spôsoby používané v súčasnosti.

Podarilo sa nám úspešne implementovať algoritmus v jazyku c#, ktorý spracuje textové časti e-mailových správ a extrahuje slovné n-gramy a tieto n-gramy porovná so zadanou správou. Pri porovnaní n-gramov sme použili hašovaciu tabuľku vstavanú v .NET frameworku. Použitím hašovacej tabuľky sme zaistili odstránenie duplicitných n-gramov. Databáza spamu ako aj výsledky detekcie sú uložené v xml súboroch. Na základe výsledkov algoritmu usudzujeme, že použitie n-gramov na detekciu spamu nie je najlepšou voľbou pre detekciu spamu. Pre úspešnú detekciu je potrebná veľká databáza spamových e-mailov. Výhodou je, že náš algoritmus nedetekoval žiadnu zhodu s databázou pri testovaní normálnych správ a mal tak nulovú false possitive. Momentálne existujú oveľa úspešnejšie metódy detekcie spamu ako napríklad Bayesov filter či Greylisting, ktorých úspešnosť je nad 99%, respektíve 95%. Náš algoritmus na extrakciu a následnú detekciu spamu dosahoval úspešnosť približne 60% pri delení textu na 4-gramy a 20-40% pri rozdelí textu na 7-gramy.

Do budúcna by bolo zaujímavé algoritmus rozšíriť o detekciu jazyka, prípadne zamerať sa na spam len v jednom jazyku, vynechať posledné znaky slov a nahradiť ho jednotným znakom pre odstránenie slov, ktoré vznikli strojovým prekladom. Jedná sa hlavne o slovenský a český jazyk, pri iných jazykoch nie je tak viditeľný strojový preklad správ. ďalším zlepšením algoritmu by mohlo byť sledovanie početnosti n-gramov.

Tomáš Tománek

11 Reference

[1] NAGAO, Makoto a Shinsuke MORI *A New Method of N-gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese.* [online]. 1994 [cit. 2013-12-19]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.8416&rep=rep1&type=pdf

[2] *Lempel-Ziv-Welch.*In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-19]. Dostupné z: http://en.wikipedia.org/wiki/Lempel-Ziv-Welch

[3] HAKAN, Ceylan a Mihalcea RADA *An Efficient Indexer for Large N-Gram Corpora.* In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations (HLT '11) [online]. [cit. 2013-12-20]. Dostupné z: http://dl.acm.org/citation.cfm?id=2002440.2002458

[4] SEDGEWICK, Robert. *Algorithms.*Reading, Mass.: Addison-Wesley, c1983, viii, 551 p. ISBN ISBN 0-201-06672-6,. Dostupné z: http://algs4.cs.princeton.edu/home/

[5] ROBENEK, Daniel, Daniel PLATOŠ a Václav SNÁŠEL *Efficient in-memory data structures for n-grams indexing.* [online]. [cit. 2013-12-20]. Dostupné z: http://ceur-ws.org/Vol-971/paper21.pdf

[6] BJORNSTRUP, Jorgen *Sorting and Searching using Hybrid AVL-Trees* [online]. 1998 [cit. 2013-12-19]. ISBN 0906-6233. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.5088&rep=rep1&type=pdf

[7] MANBER, Udi a Gene MYERS *Sufix arrays: A new method for on-line string searches.* [online]. 1991 [cit. 2013-12-20]. Dostupné z: http://webglimpse.net/pubs/suffix.pdf

[8] *Suffix tree* In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-20]. Dostupné z: http://en.wikipedia.org/wiki/Suffix_tree

[9] ZOBEL, Justin a Alistair MOFFAT *Inverted Files for Text Search Engines* [online]. 1996 [cit. 2013-12-20]. Dostupné z: http://ww2.cs.mu.oz.au/ jz/fulltext/compsurv06.pdf

[10] HALAŠ , Marián *Analýza vybraných metód a algoritmov dolovania v dátach.* In: Slovenská technická univerzita v Bratislave [online]. 2009 [cit. 2013-12-19]. Dostupné z: http://www2.fiit.stuba.sk/ kapustik/ZS/Clanky0910/halas/index.html

[11] PANTEL, Patrick a Dekang LIN. *SpamCop: A Spam Classification & Organization Program.* [online]. 1998 [cit. 2013-11-19]. DOI: AAAI Technical Report WS-98-05. Dostupné z: http://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-017.pdf

[12] ADÁMEK, Martin. *Spam: jak nepřivolávat, nepřijímat a nerozesílat nevyžádanou poštu.*, 1. vyd. Praha: Grada, 2009, 166 s. ISBN 978-80-247-2638-0.

[13] QUIGLEY, Robert. *Today in History: The First Spam Email Ever Sent.* In: ABRAMS, Dan. Geekosystem [online]. 2010 [cit. 2013-11-19]. Dostupné z: http://www.geekosystem.com/first-spam-email/

[14] SAWERS, Paul. *The origin of the word "spam".* In: [online]. [cit. 2014-03-06]. Dostupné z: http://www.thegoodword.co.uk/2010/09/20/theorigin-of-the-word-spam/.

[15] U.S. Department of state. *U.S. Relations With Nigeria.* BUREAU OF AFRICAN AFFAIRS [online]. 2013 [cit. 2013-11-19]. Dostupné z: http://www.state.gov/r/pa/ei/bgn/2836.htm

[16] MIKKELSON, Barbara. *Nigerian Scam* Snopes.com: Urban Legends Reference Pages [online]. 2010 [cit. 2013-11-19]. Dostupné z: http://www.snopes.com/fraud/advancefee/ nigeria.asp

[17] JAMES, Lance. *Phishing bez záhad* 1. vyd. Praha: Grada, 2007, 281 s. ISBN 978-80-247-1766-1.

[18] DOČEKAL, Daniel. *Rizika sociálních sítí a Webu 3.0 v praxi.* In: Lupa.cz - server o českém Internetu [online]. 2010 [cit. 2013-11-19]. Dostupné z: http://www.lupa.cz/clanky/rizika-socialnich-siti-a-webu-3-0-v-praxi/

[19] ICQ LLC. *Spam.* [online]. [cit. 2013-11-19]. Dostupné z: http://www.icq.com/support/security/spam.html

[20] LUO, Qui, Bin LIU, YAN a Zhongyue HE. *Design and Implement a Rule-Based Spam Filtering System Using Neural Network.* [online]. 2011 [cit. 2013-11-18]. DOI: 2011 International Conference on Computational and. Dostupné z: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6086218&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6085643%2F6086119%2F06086218.pdf%3Farnumber%3D6086218

[21] DIBENEDETTO, Steve, Kaustubh GADKARI, Nicholas DIEL, Andrea STEINER, Dan MASSEY a Christos PAPADOPOULOS. *Fingerprinting Custom Botnet Protocol Stacks.* Department of Computer Science Colorado State University [online]. [cit. 2013-11-21]. Dostupné z: http://www.cs.colostate.edu/ christos/papers/10npsec.pdf

[22] RAMACHANDRAN, Anirudh, Nick FEAMSTER a Santosh VEMPALA. *Filtering spam with behavioral blacklisting.* Proceedings of the 14th ACM conference on Computer and communications security - CCS '07 [online]. New York, New York, USA: ACM Press, 2007, s. 342-351 [cit. 2013-11-21]. DOI: 10.1145/1315245.1315288. Dostupné z: http://portal.acm.org/citation.cfm?doid=1315245.1315288

[23] RAMACHANDRAN, Anirudh, Shuang HAO, Hitesh KHANDELWA, Nick FEAMSTER a Santosh VEMPALA. *A Dynamic Reputation Service for Spotting Spammers.* School of Computer Science, Georgia Tech [online]. 2008 [cit. 2013-11-22]. Dostupné z: http://www.hiteshkhandelwal.com/projects/spam/spam_nsdi.pdf

[24] SILICON.DK APS. *Greylisting.org - a great weapon against spammers* [online]. [cit. 2013-11-22]. Dostupné z: http://www.greylisting.org/

[25] SILICON.DK APS. *Whitelisting*. Greylisting.org - a great weapon against spammers [online]. [cit. 2013-11-22]. Dostupné z: [online]. [cit. 2013-11-22]. Dostupné z: http://www.greylisting.org/

[26] *Greylisting.* In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-11-22]. Dostupné z: http://de.wikipedia.org/wiki/Greylisting

[27] METSIS, Vangelis, Ion ANDROUTSOPOULOS a Georgios PALIOURAS. *Spam Filtering with Naive Bayes – Which Naive Bayes?* Third Conference on Email and Anti-Spam [online]. 2006 [cit. 2013-11-22]. Dostupné z: http://classes.soe.ucsc.edu/cmps242/Fall09/lect/12/CEAS2006_corrected-naiveBayesSpam.pdf

[28] *Sender Policy Framework* [online]. 2010 [cit. 2013-11-22]. Dostupné z: http://www.openspf.org/Introduction

[29] *DomainKeys Identified Mail (DKIM)* [online]. 2007 [cit. 2013-11-22]. Dostupné z: http://dkim.org/

[30] SINČÁK, Peter a Gabriela ANDREJKOVÁ. *Neurónové siete I. (Inžiersky prístup)* FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS TECHNICAL UNIVERSITY IN KOŠICE, [online]. 1996 [cit. 2013-11-22]. Dostupné z: http://neuron-ai.tuke.sk/cig/source/publications/books/NS1/html/index.html

[31] YOUNG, Steve. *The HTK Book (for HTK Version 3.4.), 3.2 ed. Microsoft Corporation* MICROSOFT CORPORATION, Cambridge University Engineering Department., Dostupné z: http://stembep.wz.cz/!papers/ESSP05/essp15.pdf

[32] SNÁŠEL, Václav. *Znalosti 2008* Bratislava: Vydavateľstvo Slovenskej technickej univerzity, 2008, 2008. ISBN 978-80-227-2827-0.

[33] KLARLUND, Nils a Michael RILEY. *Word n-grams for cluster keyboards* TextEntry '03 Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods [online]. 2003, č. 7, s. 51-58 [cit. 2013-11-19]. DOI: Association for Computational Linguistics. Dostupné z: http://www.aclweb.org/anthology-new/W/W03/W03-2507.pdf

[34] ČEŠKA, Zdeněk. *Využití n-gramů pro odhalování plagiátů*. [online]. [cit. 2013-11-19]. Dostupné z: http://textmining.zcu.cz/publications/NGramyProPlagiaty-ITAT2007-Czech.pdf

[35] BASILE, Chiara, Dario BENEDETTO, Emanuele CAGLIOTI, Mirko DEGLI ESPOSTI a Giampaolo CRISTADORO. *A plagiarism detection procedure in three steps: selection, matches and "squares"*.[online]. [cit. 2013-11-21]. Dostupné z: www.ceur-ws.org/Vol-502/paper3.pdf

[36] *Music Ngram Viewer* [online]. 2011 [cit. 2014-01-17]. Dostupné z: http://www.peachnote.com/

[37] LIANG, Wang., *Segmenting DNA sequence into "words"based on statistical language model.* [online]. 2012 [cit. 2014-01-17]. Dostupné z: http://precedings.nature.com/documents/6939/version/1

[38] MILLINGTON, Ian a John David FUNGE, *Artificial intelligence for games.* 2nd ed. Burlington, MA: Morgan Kaufmann/Elsevier, c2009, xxiii, 870 p. ISBN 01-237-4731-7.

[39] CAVNAR, William B. a John M. TRENKLE. *N-Gram-Based Text Categorization* Proceedings of SDAIR- 94 [online]. 1994 [cit. 2013-11-21]. DOI: 3rd Annual Symposium on Document Analysis and Info. Dostupné z: www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.3248&rep=rep1&type=pdf

[40] *Limilabs — .NET components and application development* [online]. 2003-2013 [cit. 2013-12-30]. Dostupné z: http://www.limilabs.com/

[41] KRÁTKÝ, Michal, Radim BAČA, Jiří WALDER, Jiří DVORSKÝ, Peter CHOVANEC a David BEDNÁŘ., *Index-Based N-gram Extraction from Large Document Collections.*, [online]. 2012 [cit. 2014-01-31]. Dostupné z: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6093324&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6093324

[42] MILLINGTON, Ian. *Artificial intelligence for games.* Morgan Kaufmann: Elsevier, c2006, xxxv, 856 p. ISBN 01-237-3661-7.